Perturbation Resilience and Superiorization of Iterative Algorithms

Y Censor[1], **R Davidi**[2] **and G T Herman**[2]

[1]Department of Mathematics, University of Haifa, Mount Carmel, Haifa 31905, Israel

[2]Department of Computer Science, Graduate Center, City University of New York, New York, NY 10016, USA

E-mail: [1]`yair@math.haifa.ac.il`

Abstract. Iterative algorithms aimed at solving some problems are discussed. For certain problems, such as finding a common point in the intersection of a finite number of convex sets, there often exist iterative algorithms that impose very little demand on computer resources. For other problems, such as finding that point in the intersection at which the value of a given function is optimal, algorithms tend to need more computer memory and longer execution time. A methodology is presented whose aim is to produce automatically for an iterative algorithm of the first kind a "superiorized version" of it that retains its computational efficiency but nevertheless goes a long way towards solving an optimization problem. This is possible to do if the original algorithm is "perturbation resilient," which is shown to be the case for various projection algorithms for solving the consistent convex feasibility problem. The superiorized versions of such algorithms use perturbations that drive the process in the direction of the optimizer of the given function. After presenting these intuitive ideas in a precise mathematical form, they are illustrated in image reconstruction from projections for two different projection algorithms superiorized for the function whose value is the total variation of the image.

Keywords: iterative algorithms, convex feasibility problem, superiorization, perturbation resilience, projection methods

AMS classification scheme numbers: 65Y20, 68W25, 90C06, 90C25, 68U10

Submitted to: *Inverse Problems*

1. Introduction

We first motivate and describe our ideas in a not fully general context, in which superiorization is envisioned as lying in-between the methodologies of optimization and of feasibility seeking. With a feasible solution one settles for a point that just fulfills a set of constraints, whereas solving a constrained optimization problem calls for finding a feasible point that optimizes a given objective function. Generally speaking, optimization is logically and computationally a more demanding task than that of finding just any feasible point. We show that, without employing an optimization algorithm, it is possible to use certain iterative methods, designed for (the less demanding) feasibility problems, in a way that will steer the iterates toward a point that is *superior,* but not necessarily optimal, in a well-defined sense. The advantage of superiorization is that it allows us to solve significant problems by using powerful feasibility seeking methods, see, e.g., [10] and references therein, and reach a *superior feasible point* without resorting to optimization techniques. We now explain this with more details.

Many significant real-world problems are modeled by constraints that force the sought-after solution point to fulfill conditions imposed by the physical nature of the problem. Such a modeling approach often leads to a *convex feasibility problem* of the form

$$\text{find } \boldsymbol{x}^* \in C = \bigcap_{i=1}^{I} C_i, \tag{1}$$

where the sets $C_i \subseteq \mathbb{R}^J$ are closed convex subsets of the Euclidean space $\mathbb{R}^J$, see [2, 9, 16] or [15, Chapter 5] for this broad topic. In many real-world problems the underlying system is very large (huge values of I and J) and often very sparse. In these circumstances *projection methods* have proved to be effective. They are iterative algorithms that use projections onto sets while relying on the general principle that when a family of closed and convex sets is present, then projections onto the individual sets are easier to perform than projections onto other sets, such as their intersection as in (1), that are derived from them.

Projection methods can have various algorithmic structures (some of which are particularly suitable for parallel computing) and they also possess desirable convergence properties and good initial behavior patterns [2, 15, 17, 18, 19, 27, 32]. The main advantage of projection methods, which makes them successful in real-world applications, is computational. They commonly have the ability to handle huge-size problems of dimensions beyond which more sophisticated methods cease to be efficient or even applicable due to memory requirements. (For a justification of this claim see the various examples provided in [10].) This is so because the building bricks of a projection algorithm (which are the projections onto the given individual sets) are easy to perform, and because the algorithmic structure is either sequential or simultaneous, or in-between, as in the block-iterative projection methods or in the more recently invented string-averaging projection methods. The number of sets used simultaneously in each iteration in block-iterative methods and the number and lengths of strings used in each iteration in string-averaging methods are variable, which provides great flexibility in matching the implementation of the algorithm with the parallel architecture at hand; for block-iterative methods see, e.g., [1, 3, 5, 12, 19, 23, 25, 26, 29, 30, 31] and for string-averaging methods see, e.g., [4, 6, 11, 13, 14, 22, 31, 33].

The key to superiorization is our recent discovery [6, 23, 28] that two principal prototypical algorithmic schemes of projection methods: string-averaging projections (SAP) and block-iterative projections (BIP), which include as special cases a variety of projection methods for the convex feasibility problem, are bounded perturbations resilient in the sense that the convergence of sequences generated by them continues to hold even if the iterates are perturbed in every iteration. We harness this resilience to bounded perturbations to steer the iterates to not just any feasible point but to a superior (in a well-defined sense) feasible point of (1).

Our motivation is the desire to create a new methodology that will significantly improve methods for the solution of inverse problems in image reconstruction from projections, intensity-modulated radiation/proton therapy (IMRT/IMPT) and in other real-world problems such as electron microscopy (EM). Our work [6, 23], as well as the examples given below, indicate that our objective is achievable and show how algorithms can incorporate perturbations in order to perform superiorization.

The superiorization methodology has in fact broader applicability than what has been discussed until now and its mathematical specification in the next section reflects this. However, all our specific examples will be chosen from the field that we used as our motivation in this introductory section.

2. Specification of the superiorization methodology

The superiorization principle relies on the bounded perturbation resilience of algorithms. Therefore we define this notion next in a general setting within $\mathbb{R}^J$.

We introduce the notion of a *problem structure* $\langle \mathbb{T}, \mathcal{P}r \rangle$, where $\mathbb{T}$ is a nonempty *problem set* and $\mathcal{P}r$ is a function on $\mathbb{T}$ such that, for all $T \in \mathbb{T}$, $\mathcal{P}r_T : \mathbb{R}^J \to \mathbb{R}_+$, where $\mathbb{R}_+$ is the set of nonnegative real numbers. Intuitively we think of $\mathcal{P}r_T(\boldsymbol{x})$ as a measure of how "far" $\boldsymbol{x}$ is from being a solution of T. In fact, we call $\boldsymbol{x}$ a *solution* of T if $\mathcal{P}r_T(\boldsymbol{x}) = 0$.

For example, for the convex feasibility problem (1)

$$\mathbb{T} = \left\{ \{C_1, \ldots, C_I\} \middle| \begin{array}{l} I \text{ is a positive integer and, for } 1 \le i \le I, \\ C_i \text{ is a closed convex subset of } \mathbb{R}^J \end{array} \right\} \tag{2}$$

and

$$\mathcal{P}r_{\{C_1,\ldots,C_I\}}(\boldsymbol{x}) = \sqrt{\sum_{i=1}^{I} \left(d(\boldsymbol{x}, C_i) \right)^2}, \tag{3}$$

where $d(\boldsymbol{x}, C_i)$ is the Euclidean distance of $\boldsymbol{x}$ from the set C_i. Clearly, in this case $\boldsymbol{x}$ is a solution of $\{C_1, \ldots, C_I\}$ as defined in the previous paragraph if, and only if, $\boldsymbol{x} \in C$ as defined in (1).

Definition 1. An *algorithm* $\mathbf{P}$ for $\langle \mathbb{T}, \mathcal{P}r \rangle$ assigns to each $T \in \mathbb{T}$ an algorithmic operator $\mathbf{P}_T : \mathbb{R}^J \to \mathbb{R}^J$. $\mathbf{P}$ is said to be *bounded perturbations resilient* if, for all $T \in \mathbb{T}$, the following is the case: if the sequence $\left\{ (\mathbf{P}_T)^k \boldsymbol{x} \right\}_{k=0}^{\infty}$ converges to a solution of T for all $\boldsymbol{x} \in \mathbb{R}^J$, then any sequence $\left\{ \boldsymbol{x}^k \right\}_{k=0}^{\infty}$ of points in $\mathbb{R}^J$ also converges to a solution of T provided that, for all $k \ge 0$,

$$\boldsymbol{x}^{k+1} = \mathbf{P}_T \left(\boldsymbol{x}^k + \beta_k \boldsymbol{v}^k \right), \tag{4}$$

where $\beta_k \boldsymbol{v}^k$ are bounded perturbations, meaning that β_k are real nonnegative numbers such that $\sum_{k=0}^{\infty} \beta_k < \infty$ and the sequence $\left\{\boldsymbol{v}^k\right\}_{k=0}^{\infty}$ is bounded.

We give next specific instances of bounded perturbations resilient algorithms for solving the convex feasibility problem as in (2) and (3), from the classes of SAP and BIP methods. We do this by defining $\mathbf{P}_{\{C_1,\ldots,C_I\}}$ for an arbitrary but fixed element $\{C_1,\ldots,C_I\}$ of $\mathbb{T}$ of (2) for the different algorithms $\mathbf{P}$. For any nonempty closed convex subset M of $\mathbb{R}^J$ and any $\boldsymbol{x} \in \mathbb{R}^J$, the orthogonal projection of $\boldsymbol{x}$ onto M is the point in M that is nearest (by the Euclidean distance) to $\boldsymbol{x}$; it is denoted by $P_M \boldsymbol{x}$.

To define $\mathbf{P}_{\{C_1,\ldots,C_I\}}$ for the SAP instances, we make use of *index vectors,* which are nonempty ordered sets $t = (t_1, \ldots, t_N)$, where N is an arbitrary positive integer, whose elements t_n are in the set $\{1, ..., I\}$. For an index vector t we define the composite operator

$$P\left[t\right] = P_{C_{t_N}} \cdots P_{C_{t_1}}. \tag{5}$$

A finite set Ω of index vectors is called *fit* if, for each $i \in \{1, ..., I\}$, there exists $t = (t_1, \ldots, t_N) \in \Omega$ such that $t_n = i$ for some $n \in \{1, ..., N\}$. If Ω is a fit set of index vectors, then a function $\omega : \Omega \to \mathbb{R}_{++} = (0, \infty)$ is called a *fit weight function* if $\sum_{t\in\Omega} \omega\left(t\right) = 1$. A pair (Ω, ω) consisting of a fit set of index vectors and a fit weight function defined on it was called an *amalgamator* in [6]. For each amalgamator (Ω, ω), we define the algorithmic operator $\mathbf{P}_{\{C_1,\ldots,C_I\}} : \mathbb{R}^J \to \mathbb{R}^J$ by

$$\mathbf{P}_{\{C_1,\ldots,C_I\}} \boldsymbol{x} = \sum_{t\in\Omega} \omega\left(t\right) P\left[t\right] \boldsymbol{x}. \tag{6}$$

For this algorithmic operator we have the following bounded perturbations resilience theorem.

Theorem 1. [6, Section II] *If C of (1) is nonempty, $\{\beta_k\}_{k=0}^{\infty}$ is a sequence of nonnegative real numbers such that $\sum_{k=0}^{\infty} \beta_k < \infty$ and $\left\{\boldsymbol{v}^k\right\}_{k=0}^{\infty}$ is a bounded sequence of points in $\mathbb{R}^J$, then for any amalgamator (Ω, ω) and any $\boldsymbol{x}^0 \in \mathbb{R}^J$, the sequence $\left\{\boldsymbol{x}^k\right\}_{k=0}^{\infty}$ generated by*

$$\boldsymbol{x}^{k+1} = \mathbf{P}_{\{C_1,\ldots,C_I\}} \left(\boldsymbol{x}^k + \beta_k \boldsymbol{v}^k\right), \ \forall k \geq 0, \tag{7}$$

converges, and its limit is in C. (The statement of this theorem in [6] is for positive β_ks, but the proof given there applies to nonnegative β_ks.)

Corollary 1. *For any amalgamator (Ω, ω), the algorithm $\mathbf{P}$ defined by the algorithmic operator $\mathbf{P}_{\{C_1,\ldots,C_I\}}$ is bounded perturbations resilient.*

Proof. Assume that for $T = \{C_1, \ldots, C_I\}$ the sequence $\left\{\left(\mathbf{P}_T\right)^k \boldsymbol{x}\right\}_{k=0}^{\infty}$ converges to a solution of T for all $\boldsymbol{x} \in \mathbb{R}^J$. This implies, in particular, that C of (1) is nonempty. By Definition 1, we need to show that any sequence $\left\{\boldsymbol{x}^k\right\}_{k=0}^{\infty}$ of points in $\mathbb{R}^J$ also converges to a solution of T provided that, for all $k \geq 0$, (4) is satisfied when the $\beta_k \boldsymbol{v}^k$ are bounded perturbations. Under our assumptions, this follows from Theorem 1. □

Next we look at a member of the family of BIP methods. Considering the convex feasibility problem (1), for $1 \leq u \leq U$, let B_u be an ordered set $\left(b_{u,1}, \ldots, b_{u,|B_u|}\right)$ of elements of $\{1, \ldots, I\}$ ($|B_u|$ denotes the cardinality of B_u). We call such a B_u a *block* and define the (composite) algorithmic operator $\mathbf{Q}_{\{C_1,\ldots,C_I\}} : \mathbb{R}^J \to \mathbb{R}^J$ by

$$\mathbf{Q}_{\{C_1,\ldots,C_I\}} = Q_U \cdots Q_1, \tag{8}$$

where, for $\boldsymbol{x} \in \mathbb{R}^J$ and $1 \leq u \leq U$,

$$Q_u \boldsymbol{x} = \frac{1}{R} \sum_{i \in B_u} P_{C_i} \boldsymbol{x} + \frac{R - |B_u|}{R} \boldsymbol{x}, \tag{9}$$

and

$$R = \max \{ |B_u| \mid 1 \leq u \leq U \} . \tag{10}$$

The iterative procedure $\boldsymbol{x}^{k+1} = \mathbf{Q}_{\{C_1,\dots,C_I\}} \boldsymbol{x}^k$ is a member of the family of BIP methods. For this algorithmic operator we have the following bounded perturbations resilience theorem.

Theorem 2. **[23]** *If C of (1) is nonempty, $\{1, \dots, I\} = \bigcup_{u=1}^{U} B_u$, $\{\beta_k\}_{k=0}^{\infty}$ is a sequence of nonnegative real numbers such that $\sum_{k=0}^{\infty} \beta_k < \infty$, $\left\{\boldsymbol{v}^k\right\}_{k=0}^{\infty}$ be a bounded sequence of points in $\mathbb{R}^J$, then for any $\boldsymbol{x}^0 \in \mathbb{R}^J$, the sequence $\left\{\boldsymbol{x}^k\right\}_{k=0}^{\infty}$ generated by*

$$\boldsymbol{x}^{k+1} = \mathbf{Q}_{\{C_1,\dots,C_I\}} \left(\boldsymbol{x}^k + \beta_k \boldsymbol{v}^k \right), \ \forall k \geq 0, \tag{11}$$

converges, and its limit is in C. (This is a special case of Theorem 2 in [23] given here without a relaxation parameter. Also, that theorem is stated for positive β_ks, but the proof given there applies to nonnegative β_ks.)

Corollary 2. *The algorithm $\mathbf{Q}$ defined by the algorithmic operator $\mathbf{Q}_{\{C_1,\dots,C_I\}}$ is bounded perturbations resilient.*
Proof. Replace in the proof of Corollary 1 $\mathbf{P}$ by $\mathbf{Q}$ and Theorem 1 by Theorem 2. □

Further bounded perturbations resilience theorems are available in a Banach space setting, see [7, 8]. Thus the theory of bounded perturbations resilient algorithms already contains some solid mathematical results. As opposed to this, the superiorization theory that we present next is at the stage of being a collection of heuristic ideas, a full mathematical theory still needs to be developed. However, there are practical demonstrations of its potential usefulness; see [6, 23, 28] and the illustrations in Section 3 below.

For a problem structure $\langle \mathbb{T}, \mathcal{P}r \rangle$, $T \in \mathbb{T}$, $\varepsilon \in \mathbb{R}_{++}$ and a sequence $S = \left\{\boldsymbol{x}^k\right\}_{k=0}^{\infty}$ of points in $\mathbb{R}^J$, we use $O(T, \varepsilon, S)$ to denote the $\boldsymbol{x} \in \mathbb{R}^J$ that has the the following properties: $\mathcal{P}r_T(\boldsymbol{x}) \leq \varepsilon$ and there is a nonnegative integer K such that $\boldsymbol{x}^K = \boldsymbol{x}$ and, for all nonnegative integers $\ell < K$, $\mathcal{P}r_T\left(\boldsymbol{x}^\ell\right) > \varepsilon$. Clearly, if there is such an $\boldsymbol{x}$, then it is unique. If there is no such $\boldsymbol{x}$, then we say that $O(T, \varepsilon, S)$ is undefined. The intuition behind this definition is the following: if we think of S as the (infinite) sequence of points that is produced by an algorithm (intended for the problem T) without a termination criterion, then $O(T, \varepsilon, S)$ is the output produced by that algorithm when we add to it instructions that makes it terminate as soon as it reaches a point at which the value of $\mathcal{P}r_T$ is not greater than ε. The following result is obvious.

Lemma 1. *If $\mathcal{P}r_T$ is continuous and the sequence S converges to a solution of T, then $O(T, \varepsilon, S)$ is defined and $\mathcal{P}r_T(O(T, \varepsilon, S)) \leq \varepsilon$.*

Given an algorithm $\mathbf{P}$ for a problem structure $\langle \mathbb{T}, \mathcal{P}r \rangle$, a $T \in \mathbb{T}$ and an $\bar{\boldsymbol{x}} \in \mathbb{R}^J$, let $R(T, \bar{\boldsymbol{x}}) = \left\{ (\mathbf{P}_T)^k \boldsymbol{x} \right\}_{k=0}^{\infty}$. For a function $\phi : \mathbb{R}^J \to \mathbb{R}$, the *superiorization methodology* should provide us with an algorithm that produces a sequence $S(T, \bar{\boldsymbol{x}}, \phi) = \left\{\boldsymbol{x}^k\right\}_{k=0}^{\infty}$, such that for any $\varepsilon \in \mathbb{R}_{++}$ and $\bar{\boldsymbol{x}} \in \mathbb{R}^J$ for which

$\mathcal{P}r_T(\bar{\boldsymbol{x}}) > \varepsilon$ and $O(T, \varepsilon, R(T, \bar{\boldsymbol{x}}))$ is defined, $O(T, \varepsilon, S(T, \bar{\boldsymbol{x}}, \phi))$ is also defined and $\phi(O(T, \varepsilon, S(T, \bar{\boldsymbol{x}}, \phi))) < \phi(O(T, \varepsilon, R(T, \bar{\boldsymbol{x}})))$. This is of course too ambitious in its full generality and so here we analyze only a special case, but one that is still quite general. We now list our assumptions for the special case for which we discuss details of the superiorization methodology.

Assumptions

(i) $\langle \mathbb{T}, \mathcal{P}r \rangle$ is a problem structure such that $\mathcal{P}r_T$ is continuous for all $T \in \mathbb{T}$.

(ii) $\mathbf{P}$ is a bounded perturbation resilient algorithm for $\langle \mathbb{T}, \mathcal{P}r \rangle$ such that, for all $T \in \mathbb{T}$, $\mathbf{P}_T$ is continuous and, if $\boldsymbol{x}$ is not a solution of T, then $\mathcal{P}r_T(\mathbf{P}_T \boldsymbol{x}) < \mathcal{P}r_T(\boldsymbol{x})$.

(iii) ϕ is a convex function.

We now describe, under these assumptions, the algorithm to produce the sequence $S(T, \bar{\boldsymbol{x}}, \phi) = \{\boldsymbol{x}^k\}_{k=0}^{\infty}$.

The algorithm assumes that we have available a summable sequence $\{\gamma_\ell\}_{\ell=0}^{\infty}$ of positive real numbers. It is easy to generate such sequences; e.g., we can use $\gamma_\ell = a^\ell$, where $0 < a < 1$. The algorithm generates, simultaneously with the sequence $\{\boldsymbol{x}^k\}_{k=0}^{\infty}$, sequences $\{\boldsymbol{v}^k\}_{k=0}^{\infty}$ and $\{\beta_k\}_{k=0}^{\infty}$. The latter will be generated as a subsequence of $\{\gamma_\ell\}_{\ell=0}^{\infty}$. Clearly, the resulting sequence $\{\beta_k\}_{k=0}^{\infty}$ of positive real numbers will be summable. We first specify the algorithm and then discuss it. The algorithm depends on the specified $\bar{\boldsymbol{x}}$, ϕ, $\{\gamma_\ell\}_{\ell=0}^{\infty}$, $\mathcal{P}r_T$ and $\mathbf{P}_T$. It makes use of a logical variable called *continue* and also of the concept of a subgradient of the convex function ϕ.

Superiorized Version of Algorithm P

(i) **set** $k = 0$
(ii) **set** $\boldsymbol{x}^k = \bar{\boldsymbol{x}}$
(iii) **set** $\ell = 0$
(iv) **repeat**
(v)   **set** $\boldsymbol{g}$ to a subgradient of ϕ at $\boldsymbol{x}^k$
(vi)   **if** $\|\boldsymbol{g}\| > 0$
(vii)   **then set** $\boldsymbol{v}^k = -\boldsymbol{g}/\|\boldsymbol{g}\|$
(viii)   **else set** $\boldsymbol{v}^k = \boldsymbol{g}$
(ix)   **set** *continue* = *true*
(x)   **while** *continue*
(xi)   **set** $\beta_k = \gamma_\ell$
(xii)   **set** $\boldsymbol{y} = \boldsymbol{x}^k + \beta_k \boldsymbol{v}^k$
(xiii)   **if** $\phi(\boldsymbol{y}) \le \phi(\boldsymbol{x}^k)$ **and** $\mathcal{P}r_T(\mathbf{P}_T \boldsymbol{y}) < \mathcal{P}r_T(\boldsymbol{x}^k)$ **then**
(xiv)   **set** $\boldsymbol{x}^{k+1} = \mathbf{P}_T \boldsymbol{y}$
(xv)   **set** *continue* = *false*
(xvi)   **set** $\ell = \ell + 1$
(xvii)   **set** $k = k + 1$

Sometimes it is useful to emphasize the function ϕ for which we are superiorizing, in which case we refer to the algorithm above as the *ϕ-superiorized version of algorithm* $\mathbf{P}$. It is important to bear in mind that the sequence S produced by the algorithm depends also on the initial point $\bar{\boldsymbol{x}}$, the selection of the subgradient in Line (v) of the algorithm, the summable sequence $\{\gamma_\ell\}_{\ell=0}^{\infty}$, and the problem T. In addition, the output $O(T,\varepsilon,S)$ of the algorithm depends on the stopping criterion ε.

Theorem 3. *Under the Assumptions listed above, the Superiorized Version of Algorithm* $\mathbf{P}$ *will produce a sequence* $S(T,\bar{\boldsymbol{x}},\phi)$ *of points in* $\mathbb{R}^J$ *that either contains a solution of* T *or is infinite. In the latter case, if the sequence* $\left\{(\mathbf{P}_T)^k \boldsymbol{x}\right\}_{k=0}^{\infty}$ *converges to a solution of* T *for all* $\boldsymbol{x}\in\mathbb{R}^J$*, then, for any* $\varepsilon\in\mathbb{R}_{++}$*,* $O(T,\varepsilon,S(T,\bar{\boldsymbol{x}},\phi))$ *is defined and* $\phi(O(T,\varepsilon,S(T,\bar{\boldsymbol{x}},\phi)))\le\varepsilon$*.*

Proof. Assume that the sequence $S(T,\bar{\boldsymbol{x}},\phi)$ produced by the Superiorized Version of Algorithm $\mathbf{P}$ dos not contain a solution of T. We first show that in this case the algorithm generates an infinite sequence $\left\{\boldsymbol{x}^k\right\}_{k=0}^{\infty}$. This is equivalent to saying that, for any $\boldsymbol{x}^k$ that has been generated already, the condition in Line (xiii) of the algorithm will be satisfied sooner or later (and hence $\boldsymbol{x}^{k+1}$ will be generated). This needs to happen, because as long as the condition is not satisfied we keep resetting (in Line (xi)) the value of β_k to γ_ℓ, with ever increasing values of ℓ. However, $\{\gamma_\ell\}_{\ell=0}^{\infty}$ is a summable sequence of positive real numbers, and so γ_ℓ is guaranteed to be arbitrarily small if ℓ is sufficiently large. Since $\boldsymbol{v}^k$ is either a unit vector in the direction of the negative subgradient of the convex function ϕ at $\boldsymbol{x}^k$ or is the zero vector (see Lines (v)–(viii)), $\phi\left(\boldsymbol{x}^k+\beta_k\boldsymbol{v}^k\right)\le\phi\left(\boldsymbol{x}^k\right)$ must be satisfied if the positive number β_k is small enough. Also, since $\mathcal{P}r_T\left(\mathbf{P}_T\boldsymbol{x}^k\right)<\mathcal{P}r_T\left(\boldsymbol{x}^k\right)$ and $\mathbf{P}_T$ and $\mathcal{P}r_T$ are continuous (Assumptions (ii) and (i), respectively), we also have that $\mathcal{P}r_T\left(\mathbf{P}_T\left(\boldsymbol{x}^k+\beta_k\boldsymbol{v}^k\right)\right)<\mathcal{P}r_T\left(\boldsymbol{x}^k\right)$ if β_k is small enough. This completes the proof that the condition in Line (xiii) of the algorithm will be satisfied and so the algorithm will generate an infinite sequence $S(T,\bar{\boldsymbol{x}},\phi)$. Observing that we have already demonstrated that the $\beta_k\boldsymbol{v}^k$ are bounded perturbations, and comparing (4) with Lines (xii) and (xiv), we see that (by the bounded perturbation resilience of $\mathbf{P}$) the assumption that the sequence $\left\{(\mathbf{P}_T)^k \boldsymbol{x}\right\}_{k=0}^{\infty}$ converges to a solution of T for all $\boldsymbol{x}\in\mathbb{R}^J$ implies that $S(T,\bar{\boldsymbol{x}},\phi))$ also converges to a solution of T. Thus, applying Lemma 1 we obtain the final claim of the theorem. □

Unfortunately, this theorem does not go far enough. To demonstrate that a methodology leads to superiorization we should be proving (under some assumptions) a result like $\phi(O(T,\varepsilon,S(T,\bar{\boldsymbol{x}},\phi)))<\phi(O(T,\varepsilon,R(T,\bar{\boldsymbol{x}})))$ in place of the weaker result at the end of the statement of the theorem. Currently we do not have any such proofs and so we are restricted to providing practical demonstrations that our methodology leads to superiorization in the desired sense. In the next section we provide such demonstrations for the Superiorized Version of Algorithm $\mathbf{P}$, for two different $\mathbf{P}$s.

3. Illustrations of the superiorization methodology

We illustrate the superiorization methodology on a problem of reconstructing a head cross-section (based on Figure 4.6(a) of [27]) from its projections using both an SAP and a BIP algorithm. (All the computational work reported in this section was done

Table 1: Values of TV for the outputs of the various algorithms. The second column is for the superiorized versions and the third column is for the original versions.

Algorithm	$\phi(O(T,\varepsilon,S(T,\bar{\boldsymbol{x}},\phi)))$	$\phi(O(T,\varepsilon,R(T,\bar{\boldsymbol{x}})))$
Variant of ART	441.50	1,296.44
Variant of BIP	444.15	1,286.44

using SNARK09 [24]; the phantom, the data, the reconstructions and displays were all generated within this same framework.) Figure 1(a) shows a 243×243 digitization of the head phantom with $J = 59,049$ pixels. An $\boldsymbol{x} \in \mathbb{R}^J$ is interpreted as a vector of pixel values, whose components represent the average X-ray linear attenuation coefficients (measured per centimeter) within the 59,049 pixels. Each pixel is of size 0.0752×0.0752 (measured in centimeters). The pixel values range from 0 to 0.5639. For display purposes, any value below 0.204 is shown as black (gray value 0) and any value above 0.21675 is shown as white (gray value 255), with a linear mapping of the pixel values into gray values in between (the same convention is used in displaying reconstructed images in Figures 1(b)-(e)).

Data were collected by calculating line integrals across the digitized image for 82 sets of equally spaced parallel lines, with $I = 25,452$ lines in total. Each data item determines a hyperplane in $\mathbb{R}^J$. Since the digitized phantom lies in the intersection of all the hyperplanes, we have here an instance of the convex feasibility problem with a nonempty C, satisfying the first condition of the statements of Theorems 1 and 2.

For our illustration, we chose the SAP algorithm $\mathbf{P}_{\{C_1,\ldots,C_I\}}$ as determined by (5)-(6) with $\Omega = \{(1,\ldots,I)\}$ and $\omega(1,\ldots,I) = 1$. This is a classical method that in tomography would be considered a variant of the algebraic reconstruction techniques (ART) [27, Chapter 11]. For the BIP algorithm we chose $\mathbf{Q}_{\{\mathbf{C_1},\ldots,\mathbf{C_I}\}}$ as determined by (8)-(10) with $U = 82$ and each block corresponding to one of the 82 sets of parallel lines along which the data are collected.

The function ϕ for which we superiorized is defined so that, for any $\boldsymbol{x} \in \mathbb{R}^J$, $\phi(\boldsymbol{x})$ is the *total variation* (TV) of the corresponding 243×243 image. If the pixel values of this image are $q_{g,h}$, then the value of the TV is defined to be

$$\sum_{g=1}^{242}\sum_{h=1}^{242}\sqrt{\left(q_{g+1,h}-q_{g,h}\right)^2+\left(q_{g,h+1}-q_{g,h}\right)^2}. \tag{12}$$

For the TV-Superiorized Versions of the Algorithms $\mathbf{P}_{\{C_1,\ldots,C_I\}}$ and $\mathbf{Q}_{\{\mathbf{C_1},\ldots,\mathbf{C_I}\}}$ of the previous paragraph we selected $\bar{\boldsymbol{x}}$ to be the origin (the vector of all zeros) and $\gamma_\ell = 0.999^\ell$. Also , we set $\varepsilon = 0.01$ for the stopping criterion, which is small compared to the $\mathcal{P}r_T$ of the initial point ($\mathcal{P}r_T(\bar{\boldsymbol{x}}) = 330.208$).

For each of the four algorithms ($\mathbf{P}_{\{C_1,\ldots,C_I\}}$, $\mathbf{Q}_{\{C_1,\ldots,C_I\}}$ and their TV-superiorized versions), the sequence S that is produced by it is such that the output $O(T,\varepsilon,S)$ is defined; see Figures 1(b)-(e) for the images that correspond to these outputs. Clearly, the superiorized reconstructions in Figures 1(c) and (e) are visually superior to their not superiorized versions in Figure 1(b) and (d), respectively. More importantly from the point of view of our theory, consider Table 1. As stated in the last paragraph of the previous section, we would like to have that $\phi(O(T,\varepsilon,S(T,\bar{\boldsymbol{x}},\phi))) < \phi(O(T,\varepsilon,R(T,\bar{\boldsymbol{x}})))$. While we are not able to prove that this is the case in general, Table 1 clearly shows it to be the case for the two algorithms discussed in this section.

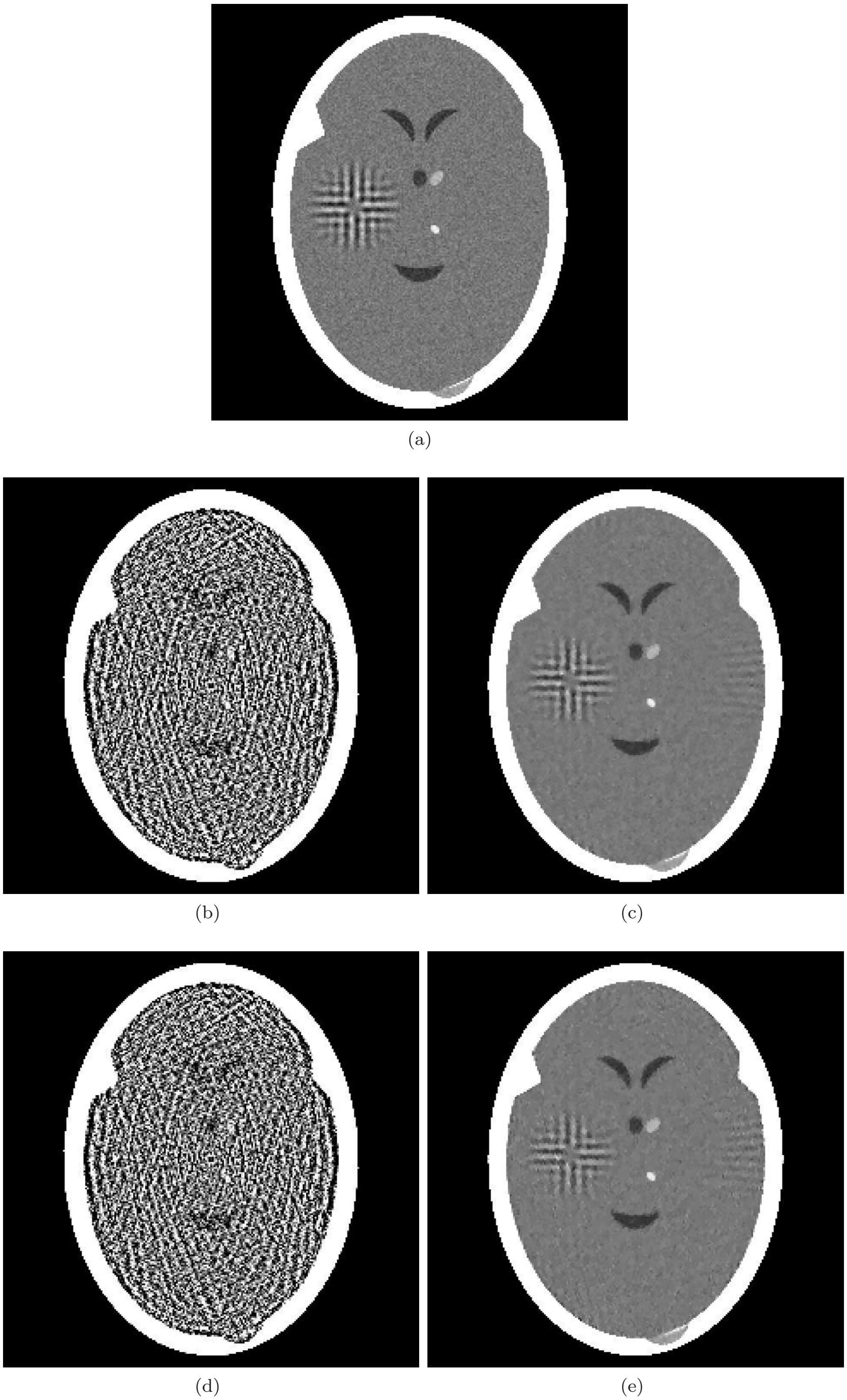

Figure 1: A head phantom (a) and its reconstructions from underdetermined consistent data obtained for 82 views using: (b) a variant of ART, (c) TV-superiorized version of the same variant of ART, (d) a block-iterative projection method, and (e) TV-superiorized version of the same block-iterative projection method. The same initial point and stopping criterion were used in all cases; see the text for details.

A final important point that is illustrated by the experiments in this section is that, from the practical point of view, TV-superiorization is as useful as TV-optimization. This is because a realistic phantom, such as the one in Figure 1(a), is unlikely to be TV-minimizing subject to the constraints provided by the measurements. In fact, the TV value of our phantom is 450.53, which is larger than that for either of the TV-superiorized reconstructions in the second column of Table 1. While an optimization method should be able to find an image with a lower TV value, there is no practical point for doing that. The underlying aim of what we are doing is to estimate the phantom from the data and producing an image whose TV value is further from the TV value of the phantom than that of our superiorized reconstructions is unlikely to be helpful towards achieving this aim.

4. Discussion and conclusions

Stability of algorithms under perturbations is generally studied in numerical analysis with the aim of proving that an algorithm is stable so that it can "endure" all kinds of imperfections in the data or in the computational performance. Here we have taken a proactive approach designed to extract specific benefits from the kind of stability that we term perturbation resilience. We have been able to do this in a context that includes, but is much more general than, feasibility-optimization for intersections of convex sets.

Our premise has been that (1) there is available a bounded perturbations resilient iterative algorithm that solves efficiently certain type of problems and (2) we desire to make use of perturbations to find for these problems solutions that, according to some criterion, are superior to the ones to which we would get without employing perturbations. To accomplish this one must have a way of introducing perturbations that take into account the criterion according to which we wish to "superiorize" the solutions of the problems.

We have set forth the fundamental principle, have given some mathematical formulations and results, and have shown potential benefits (in the field of image reconstruction from projections). However, the superiorization methodology needs to be studied further from the mathematical, algorithmic and computational points of view in order to unveil its general applicability to inverse problems. As algorithms are developed and tested a dialog on algorithmic developments must be accompanied by mathematical validation and applications to simulated and real data from various relevant fields of applications.

Validating the concept means proving precise statements about the behavior of iterates $\left\{\boldsymbol{x}^k\right\}_{k=0}^{\infty}$ generated by the superiorized versions of algorithms. Under what conditions do they converge? Can their limit points be characterized? How would different choices of the perturbation coefficients β_k and the perturbation vectors $\boldsymbol{v}^k$ affect the superiorization process? Can different schemes for generating the β_ks be developed, implemented, investigated? Enlarging the arsenal of bounded perturbation resilience algorithms means generalizing existing proofs of bounded perturbations resiliency of algorithms and developing new theories that will bring more algorithms into the family of bounded perturbations resilient algorithms. Further developments should include the problem of finding a common fixed point of a family of operators (a direct generalization of the convex feasibility problem) and studying the behavior of superiorization algorithms in inconsistent situations when the underlying solution set is empty. Thus we view the material in this paper as only an initial step in a

promising new field of endeavor for solving inverse problems.

Acknowledgments

This work was supported by Award Number R01HL070472 from the National Heart, Lung, And Blood Institute. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Heart, Lung, And Blood Institute or the National Institutes of Health.

References

[1] Aharoni R and Censor Y 1989 Block-iterative projection methods for parallel computation of solutions to convex feasibility problems *Linear Algebra Appl.* **120**, 165–75
[2] Bauschke HH and Borwein JM 1996 On projection algorithms for solving convex feasibility problems *SIAM Rev.* **38** 367–426
[3] Bauschke HH, Combettes PL and Kruk SG 2006 Extrapolation algorithm for affine-convex feasibility problems *Numer. Algorithms* **41** 239–74
[4] Bauschke HH, Matoušková E and Reich S 2004 Projection and proximal point methods: convergence results and counterexamples. *Nonlinear Anal.* **56** 715–38
[5] Butnariu D and Censor Y 1990 On the behavior of a block-iterative projection method for solving convex feasibility problems *Int. J. Comput Math.* **34** 79–94
[6] Butnariu D, Davidi R, Herman GT and Kazantsev IG 2007 Stable convergence behavior under summable perturbations of a class of projection methods for convex feasibility and optimization problems *IEEE J. Sel. Top. Sign. Process.* **1** 540–7
[7] Butnariu D, Reich S and Zaslavski AJ 2006 Convergence to fixed points of inexact orbits of Bregman-monotone and nonexpansive operators in Banach spaces *Fixed Point Theory and Applications* ed H F Nathansky, B G de Buen, K Goebel, W A Kirk and B Sims (Yokohama: Yokohama Publishers) pp 11–32
[8] Butnariu D, Reich S and Zaslavski AJ 2008 Stable convergence theorems for infinite products and powers of nonexpansive mappings *Numer. Func. Anal. Opt.* **29** 304–23
[9] Byrne CL 2008 *Applied Iterative Methods* (AK Peters)
[10] Censor Y, Chen W, Combettes PL, Davidi R and Herman GT 2009 On the effectiveness of projection methods for convex feasibility problems with linear inequality constraints, *Opt. Online*, http://www.optimization-online.org/DB_HTML/2009/12/2500.html
[11] Censor Y, Elfving T and Herman GT 2001 Averaging strings of sequential iterations for convex feasibility problems. *Inherently Parallel Algorithms in Feasibility and Optimization and Their Applications* ed Butnariu D, Censor Y and Reich S (Elsevier Science Publishers) pp 101–14
[12] Censor Y, Gordon D and Gordon R 2001 BICAV: A block-iterative, parallel algorithm for sparse systems with pixel-related weighting *IEEE Trans. Med. Imaging* **20** 1050–60
[13] Censor Y and Segal A 2009 On the string averaging method for sparse common fixed points problems *Int. Trans. Oper. Res.* **16** 481–94
[14] Censor Y and Tom E 2003 Convergence of string-averaging projection schemes for inconsistent convex feasibility problems *Optim. Methods Softw.* **18** 543–54
[15] Censor Y and Zenios SA 1997 *Parallel Optimization: Theory, Algorithms and Applications* (Oxford University Press)
[16] Chinneck JW 2007 *Feasibility and Infeasibility in Optimization: Algorithms and Computational Methods* (Springer)
[17] Combettes PL 1996 The convex feasibility problem in image recovery *Adv. Imag. Elec. Phys.* **95** 155–270
[18] Combettes PL 1997 Hilbertian convex feasibility problem: Convergence of projection methods *Appl. Math. Opt.* **35** 311–30
[19] Combettes PL 1997 Convex set theoretic image recovery by extrapolated iterations of parallel subgradient projections *IEEE T. Image Process.* **6** 493–506
[20] Combettes PL and Luo J 2002 An adaptive level set method for nondifferentiable constrained image recovery *IEEE T. Image Process.* **11** 1295–304
[21] Combettes PL and Pesquet JC 2004 Image restoration subject to a total variation constraint *IEEE T. Image Process.* **13** 1213–22
[22] Crombez G 2002 Finding common fixed points of strict paracontractions by averaging strings of sequential iterations *J. Nonlinear Convex Anal* **3** 345–51

[23] Davidi R, Herman GT and Censor Y 2009 Perturbation-resilient block-iterative projection methods with application to image reconstruction from projections *Int. Trans. Oper. Res.* **16** 505–24

[24] Davidi R, Herman GT and Klukowska J 2009 SNARK09: A programming system for the reconstruction of 2D images from 1D projections (http://www.snark09.com/)

[25] Eggermont PPB, Herman GT and Lent A 1981 Iterative algorithms for large partitioned linear systems, with applications to image reconstruction *Linear Algebra Appl.* **40** 37–67

[26] González-Castaño FJ, García-Palomares UM, Alba-Castro JL and Pousada-Carballo JM 2001 Fast image recovery using dynamic load balancing in parallel architectures, by means of incomplete projections *IEEE T. Image Process.* **10** 493–99

[27] Herman GT 2009 *Fundamentals of Computerized Tomography: Image Reconstruction from Projections* 2nd ed. (Springer)

[28] Herman GT and Davidi R 2008 On image reconstruction from a small number of projections *Inverse Problems* **24**:045011

[29] Kiwiel KC and Łopuch B 1997 Surrogate projection methods for finding fixed points of firmly nonexpansive mappings *SIAM J. Optim.* **7** 1084–1102

[30] Ottavy N 1988 Strong convergence of projection-like methods in Hilbert spaces *J. Optim. Theory Appl.* **56** 433–461

[31] Penfold SN, Schulte RW, Censor Y, Bashkirov V, McAllister S, Schubert KE, Rosenfeld AB (to appear) Block-iterative and string-averaging projection algorithms in proton computed tomography image reconstruction ed Censor Y, Jiang M and Wang G *Biomedical Mathematics: Promising Directions in Imaging, Therapy Planning and Inverse Problems* (Medical Physics Publishing)

[32] Pierra G 1984 Decomposition through formalization in a product space *Math. Program.* **28** 96–115

[33] Rhee H 2003 An application of the string averaging method to one-sided best simultaneous approximation *J. Korea Soc. Math. Educ. Ser. B Pure Appl. Math.* **10** 49–56